Acceleration Methods and Discrete Soliton Equations

NAGAI, Atsushi; SATSUMA, Junkichi

数理解析研究所講究録
933 巻 1995 年 44-60

Acceleration Methods and Discrete Soliton Equations

(加速法と離散型ソリトン方程式)

Atsushi NAGAI (永井 敦) †

Junkichi SATSUMA (薩摩 順吉)

Department of Mathematical Sciences, University of Tokyo

† e-mail: nagai@sat.t.u-tokyo.ac.jp

§1 Introduction

Integrable discretization of soliton equations has been in progress since 1970's[1]. Recently discrete soliton equations have attracted attention in other fields such as engineering. For example, finite, nonperiodic Toda equation appears in the field of matrix eigenvalue algorithm [2, 3]. The discrete (potential) KdV equation is nothing but one of the most popular convergence acceleration schemes, the $\varepsilon-$algorithm [4].

Our main interest in this paper is on the convergence acceleration algorithms. Let $\{S_m\}$ be a sequence of numbers which converges to S_∞. In order to find S_∞, direct calculation often requires a large amount of data. Sequences

$$S_m = 1-\frac{1}{2}+\frac{1}{3}-\cdots+\frac{(-1)^m}{m+1}, \tag{1}$$

$$S_m = 1+\frac{1}{2^2}+\frac{1}{3^2}+\cdots+\frac{1}{(m+1)^2} \tag{2}$$

are typical examples. Beside these simple cases, one has often to deal with slowly convergent sequences in the field of numerical analysis, applied mathematics, and engineering. In such cases we transform the original sequence $\{S_m\}$ into another sequence $\{T_m\}$ instead of calculating directly. If $\{T_m\}$ converges to S_∞ faster than $\{S_m\}$, that is

$$\lim_{m\to\infty}\frac{T_m-S_\infty}{S_m-S_\infty}=0, \tag{3}$$

[1] See [1], for example.

we say that the transformation $T : \{S_m\} \to \{T_m\}$ *accelerates the convergence* of the sequence $\{S_m\}$. We now have various kinds of convergence acceleration algorithms such as $\varepsilon-$algorithm [5], $\eta-$algorithm [6], $\rho-$algorithm [7, 8], BS-algorithm [9], Levin's $t-, u-$, and $v-$transformation [10], and $\theta-$algorithm [11].

The main purpose of this paper is to study acceleration methods from a different aspect, that is, the soliton theory. In §2, we introduce Bauer's $\eta-$algorithm and show its equivalence with the discrete KdV equation [1]. In §3, we look over the result by Papageorgiou et al., the equivalence between Wynn's $\varepsilon-$algorithm and the discrete potential KdV equation. In §4, we introduce a different type of algorithm, Wynn's $\rho-$algorithm. In spite of its similarity with the $\varepsilon-$algorithm, it possesses noticeably different characteristics not only as a convergence accelerator but also as a discrete soliton equation. When we respect the $\rho-$algorithm as a two-variable difference equation, its solution is represented by double Casorati determinant. We show in §5 that this fact is quite natural if we discuss the $\rho-$algorithm in relation with Thiele's interpolation formula [7]. We also present the Thiele's $\rho-$algorithm, which is one generalization of the $\rho-$algorithm, and compare its performance with the original $\rho-$algorithm. In §6, we consider the "PGR algorithms", which are the most generalized rhombus algorithms satisfying the singularity confinement condition. Concluding remarks are given in §7.

§2 The $\eta-$algorithm

In this section we show that Bauer's $\eta-$algorithm[6], which is one of the famous convergence acceleration algorithms, is equivalent to the discrete KdV equation. The $\eta-$algorithm involves a two-dimensional array called the $\eta-$table (Figure 1). The table is constructed from its first two columns. Let initial values $\eta_0^{(m)}$ and $\eta_1^{(m)}$ be

$$\eta_0^{(m)} = \infty, \ \eta_1^{(m)} = c_m \equiv \Delta S_{m-1}, \ (m = 0, 1, 2, \ldots), \ S_{-1} \equiv 0, \tag{4}$$

where Δ is the forward difference operator given by $\Delta a_k = a_{k+1} - a_k$. Then all the other elements are calculated from the following recurrence relations called the $\eta-$algorithm;

$$\left\{ \begin{array}{rcl} \eta_{2n+1}^{(m)} + \eta_{2n}^{(m)} & = & \eta_{2n}^{(m+1)} + \eta_{2n-1}^{(m+1)} \\ \dfrac{1}{\eta_{2n+2}^{(m)}} + \dfrac{1}{\eta_{2n+1}^{(m)}} & = & \dfrac{1}{\eta_{2n+1}^{(m+1)}} + \dfrac{1}{\eta_{2n}^{(m+1)}} \end{array} \right. \quad \text{(rhombus rules)}. \tag{5}$$

Equation (5) defines a transformation of a given series $c_m = \eta_1^{(m)}, m = 0, 1, 2, \ldots$ to a new series $c'_n = \eta_n^{(0)}, n = 1, 2, \ldots$ such that $\sum_{n=1}^{\infty} c'_n$ converges more rapidly to the same limit S_∞.

$$
\begin{array}{ccccccc}
 & \eta_1^{(0)} & & & & & \\
(\infty =)\eta_0^{(1)} & & \eta_2^{(0)} & & & & \\
 & \eta_1^{(1)} & & \eta_3^{(0)} & & & \\
(\infty =)\eta_0^{(2)} & & \eta_2^{(1)} & & \eta_4^{(0)} & & \\
 & \eta_1^{(2)} & & \eta_3^{(1)} & & \eta_5^{(0)} & \\
(\infty =)\eta_0^{(3)} & & \eta_2^{(2)} & & \eta_4^{(1)} & \vdots & \ddots \\
 & \eta_1^{(3)} & & \eta_3^{(2)} & \vdots & & \\
(\infty =)\eta_0^{(4)} & & \eta_2^{(3)} & \vdots & & & \\
\vdots & \eta_1^{(4)} & \vdots & & & & \\
 & \vdots & & & & &
\end{array}
$$

Figure 1: The $\eta-$table.

As a simple example we consider a slowly convergent series (1) and construct the $\eta-$table. We see from Figure 2 that the transformed series

$$1-\frac{1}{3}+\frac{1}{30}-\frac{1}{130}+\frac{1}{975}-\frac{1}{4725}+\cdots$$

converges more rapidly to $\log 2$ than the original series. While the sum of the first seven terms of the original series gives $0.7595\cdots$, that of the corresponding seven terms of the transformed series does $0.693152\cdots$.

$$
\begin{array}{cccccccc}
 & 1 & & & & & & \\
\infty & & -1/3 & & & & & \\
 & -1/2 & & 1/30 & & & & \\
\infty & & 1/5 & & -1/130 & & & \\
 & 1/3 & & -1/105 & & 1/975 & & \\
\infty & & -1/7 & & 1/350 & & -1/4725 & \\
 & -1/4 & & 1/252 & & -1/4100 & & 1/32508 \\
\infty & & 1/9 & & -1/738 & & 1/15867 & \\
 & 1/5 & & -1/495 & & 1/12505 & & \\
\infty & & -1/11 & & 1/1342 & & & \\
 & -1/6 & & 1/858 & & & & \\
\infty & & 1/13 & & & & & \\
 & 1/7 & & & & & &
\end{array}
$$

Figure 2: The $\eta-$table for $\log 2$

The quantities $\eta_n^{(m)}$ are given by the following ratios of Hankel determinants;

$$\eta_{2n+1}^{(m)} = \frac{\begin{vmatrix} c_m & \cdots & c_{m+n} \\ \vdots & & \vdots \\ c_{m+n} & \cdots & c_{m+2n} \end{vmatrix} \cdot \begin{vmatrix} c_{m+1} & \cdots & c_{m+n} \\ \vdots & & \vdots \\ c_{m+n} & \cdots & c_{m+2n-1} \end{vmatrix}}{\begin{vmatrix} \Delta c_m & \cdots & \Delta c_{m+n-1} \\ \vdots & & \vdots \\ \Delta c_{m+n-1} & \cdots & \Delta c_{m+2n-2} \end{vmatrix} \cdot \begin{vmatrix} \Delta c_{m+1} & \cdots & \Delta c_{m+n} \\ \vdots & & \vdots \\ \Delta c_{m+n} & \cdots & \Delta c_{m+2n-1} \end{vmatrix}}, \tag{6}$$

$$\eta_{2n+2}^{(m)} = \frac{\begin{vmatrix} c_m & \cdots & c_{m+n} \\ \vdots & & \vdots \\ c_{m+n} & \cdots & c_{m+2n} \end{vmatrix} \cdot \begin{vmatrix} c_{m+1} & \cdots & c_{m+n+1} \\ \vdots & & \vdots \\ c_{m+n+1} & \cdots & c_{m+2n+1} \end{vmatrix}}{\begin{vmatrix} \Delta c_m & \cdots & \Delta c_{m+n} \\ \vdots & & \vdots \\ \Delta c_{m+n} & \cdots & \Delta c_{m+2n} \end{vmatrix} \cdot \begin{vmatrix} \Delta c_{m+1} & \cdots & \Delta c_{m+n} \\ \vdots & & \vdots \\ \Delta c_{m+n} & \cdots & \Delta c_{m+2n-1} \end{vmatrix}}. \tag{7}$$

If we introduce dependent variable transformations,

$$X_{2n}^{(m)} = \frac{1}{\eta_{2n}^{(m)}}, \; X_{2n-1}^{(m)} = \eta_{2n-1}^{(m)} \tag{8}$$

the η−algorithm (5) is rewritten as

$$X_{n+1}^{(m)} - X_{n-1}^{(m+1)} = \frac{1}{X_n^{(m+1)}} - \frac{1}{X_n^{(m)}}, \tag{9}$$

which is the Hirota's discrete KdV equation [1].

§3 The ε−algorithm

In this section, following the result by Papageorgiou et al., we briefly review the equivalence the discrete potential KdV equation and the ε−algorithm, which originates with Shanks [13] and Wynn [5]. The algorithm involves a two-dimensional array called the ε−table (Figure 3). Define $\varepsilon_0^{(m)}$ and $\varepsilon_1^{(m)}$ by

$$\varepsilon_0^{(m)} = 0, \; \varepsilon_1^{(m)} = S_m \; (m = 0, 1, 2, \ldots). \tag{10}$$

Then all the other quantities obey the following rhombus rule;

$$(\varepsilon_{n+1}^{(m)} - \varepsilon_{n-1}^{(m+1)})(\varepsilon_n^{(m+1)} - \varepsilon_n^{(m)}) = 1. \tag{11}$$

$$
\begin{array}{cccccccc}
 & \varepsilon_1^{(0)} & & & & & & \\
(0=)\varepsilon_0^{(1)} & & \varepsilon_2^{(0)} & & & & & \\
 & \varepsilon_1^{(1)} & & \varepsilon_3^{(0)} & & & & \\
(0=)\varepsilon_0^{(2)} & & \varepsilon_2^{(1)} & & \varepsilon_4^{(0)} & & & \\
 & \varepsilon_1^{(2)} & & \varepsilon_3^{(1)} & & \varepsilon_5^{(0)} & & \\
(0=)\varepsilon_0^{(3)} & & \varepsilon_2^{(2)} & & \varepsilon_4^{(1)} & \vdots & \ddots & \\
 & \varepsilon_1^{(3)} & & \varepsilon_3^{(2)} & \vdots & & & \\
(0=)\varepsilon_0^{(4)} & & \varepsilon_2^{(3)} & \vdots & & & & \\
\vdots & \varepsilon_1^{(4)} & \vdots & & & & & \\
 & \vdots & & & & & &
\end{array}
$$

Figure 3: The ε−table

According as n becomes large, $\varepsilon_{2n+1}^{(m)}$ converges more rapidly to S_∞ as $m \to \infty$. On the other hand, $\varepsilon_{2n}^{(m)}$ diverges as $m \to \infty$.

It has been shown that the ε−algorithm (11) is regarded as the discrete potential KdV equation [14]. The quantities $\varepsilon_n^{(m)}$ are also given by the following ratios of Hankel determinants;

$$
\varepsilon_{2n+1}^{(m)} = \frac{\begin{vmatrix} S_m & S_{m+1} & \cdots & S_{m+n} \\ S_{m+1} & S_{m+2} & \cdots & S_{m+n+1} \\ \vdots & \vdots & & \vdots \\ S_{m+n} & S_{m+n+1} & \cdots & S_{m+2n} \end{vmatrix}}{\begin{vmatrix} \Delta^2 S_m & \Delta^2 S_{m+1} & \cdots & \Delta^2 S_{m+n-1} \\ \Delta^2 S_{m+1} & \Delta^2 S_{m+2} & \cdots & \Delta^2 S_{m+n} \\ \vdots & \vdots & & \vdots \\ \Delta^2 S_{m+n-1} & \Delta^2 S_{m+n} & \cdots & \Delta^2 S_{m+2n-2} \end{vmatrix}}, \tag{12}
$$

$$
\varepsilon_{2n+2}^{(m)} = \frac{\begin{vmatrix} \Delta^3 S_m & \Delta^3 S_{m+1} & \cdots & \Delta^3 S_{m+n-1} \\ \Delta^3 S_{m+1} & \Delta^3 S_{m+2} & \cdots & \Delta^3 S_{m+n} \\ \vdots & \vdots & & \vdots \\ \Delta^3 S_{m+n-1} & \Delta^3 S_{m+n} & \cdots & \Delta^3 S_{m+2n-2} \end{vmatrix}}{\begin{vmatrix} \Delta S_m & \Delta S_{m+1} & \cdots & \Delta S_{m+n} \\ \Delta S_{m+1} & \Delta S_{m+2} & \cdots & \Delta S_{m+n+1} \\ \vdots & \vdots & & \vdots \\ \Delta S_{m+n} & \Delta S_{m+n+1} & \cdots & \Delta S_{m+2n} \end{vmatrix}}. \tag{13}
$$

Equation (12) is called the Shanks transformation [13]. Substitution of $n = 1$ in eq. (12) gives the well-known Aitken acceleration algorithm.

We have so far observed that the $\eta-$ and the $\varepsilon-$algorithms are interpreted as the discrete KdV and the discrete potential KdV equations, respectively. Therefore, these two algorithms are the same in their performance as convergence acceleration algorithms. This equivalence can also be understood from the fact [6] that the quantities $\eta_n^{(m)}$ and $\varepsilon_n^{(m)}$ are related by

$$\eta_{2n}^{(m)} = \varepsilon_{2n+1}^{(m-1)} - \varepsilon_{2n-1}^{(m)}, \ \eta_{2n+1}^{(m)} = \varepsilon_{2n+1}^{(m)} - \varepsilon_{2n+1}^{(m-1)}. \tag{14}$$

§4 The $\rho-$algorithm

The $\rho-$algorithm is traced back to Thiele's rational interpolation [7]. It was first used as a convergence accelerator by Wynn [8]. The initial values of the algorithm are given by

$$\rho_0^{(m)} = 0, \ \rho_1^{(m)} = S_m \ (m = 0, 1, 2, \ldots), \tag{15}$$

and all the other elements fulfill the following rhombus rule;

$$(\rho_{n+1}^{(m)} - \rho_{n-1}^{(m+1)})(\rho_n^{(m+1)} - \rho_n^{(m)}) = n. \tag{16}$$

The $\rho-$algorithm is almost the same as the $\varepsilon-$algorithm except that "1" in the right hand side of eq. (11) is replaced by "n" in eq. (16). This slight change, however, yields considerable differences in various aspects between these two algorithms.

The first difference is in their performance. As one can find in ref. [15], the $\varepsilon-$algorithm accelerates exponentially or alternatively decaying sequences, while the $\rho-$algorithm does rationally decaying sequences.

The second difference is in their determinant expressions. The quantities $\varepsilon_n^{(m)}$ are given by ratios of Hankel determinants, while the quantities $\rho_n^{(m)}$ are given by [7]

$$\rho_n^{(m)} = (-1)^{[\frac{n-1}{2}]} \frac{\tilde{\tau}_n^{(m)}}{\tau_n^{(m)}}, \tag{17}$$

where $[x]$ stands for the greatest integer less than or equal to x. Moreover, the functions $\tau_n^{(m)}$ and $\tilde{\tau}_n^{(m)}$ are expressed as the following double Casorati determinants;

$$\tau_n^{(m)} = \begin{cases} u^{(m)}(k;k) & n = 2k, \\ u^{(m)}(k+1;k) & n = 2k+1, \end{cases} \tag{18}$$

$$\tilde{\tau}_n^{(m)} = \begin{cases} u^{(m)}(k+1;k-1) & n = 2k, \\ u^{(m)}(k;k+1) & n = 2k+1, \end{cases} \tag{19}$$

where

$$u^{(m)}(p;q) = \det \left[\begin{array}{cccc|cccc} 1 & m & \cdots & m^{p-1} & S_m & mS_m & \cdots & m^{q-1}S_m \\ 1 & m+1 & \cdots & (m+1)^{p-1} & S_{m+1} & (m+1)S_{m+1} & \cdots & (m+1)^{q-1}S_{m+1} \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 1 & m+p+q-1 & \cdots & (m+p+q-1)^{p-1} & S_{m+p+q-1} & (m+p+q-1)S_{m+p+q-1} & \cdots & (m+p+q-1)^{q-1}S_{m+p+q-1} \end{array} \right]. \tag{20}$$

We remark that a pair of functions $\tau_n^{(m)}$ and $\tilde{\tau}_n^{(m)}$ given by eqs. (18) and (19) satisfy bilinear equations,

$$\tau_{n+1}^{(m)}\tau_{n-1}^{(m+1)} - \tau_n^{(m)}\tilde{\tau}_n^{(m+1)} + \tau_n^{(m+1)}\tilde{\tau}_n^{(m)} = 0, \tag{21}$$

$$\tau_{n-1}^{(m+1)}\tilde{\tau}_{n+1}^{(m)} + \tau_{n+1}^{(m)}\tilde{\tau}_{n-1}^{(m+1)} - n\tau_n^{(m)}\tau_n^{(m+1)} = 0, \tag{22}$$

which are considered to be the Jacobi and the Plücker identities for determinants, respectively.

§5 Reciprocal Differences and Thiele's ρ–algorithm

In this section we extend the ρ–algorithm from the viewpoint of the τ function and compare its performance with the original ρ–algorithm (16). Before touching upon the extended version of the ρ–algorithm, we review Thiele's interpolation [7], which makes it natural that quantities $\rho_n^{(m)}$ are given by ratios of double Casorati determinants. Let the values of an unknown function $f(x)$ be given for the values $x_0, x_1, \cdots, x_n$, no two of which are equal. Then reciprocal differences $\rho_i(x_k x_{k+1} \cdots x_{k+i-1})$ are defined by

$$\rho_1(x_0) = f(x_0), \tag{23}$$

$$\rho_2(x_0x_1) = \frac{x_0 - x_1}{\rho_1(x_0) - \rho_1(x_1)}, \tag{24}$$

$$\rho_3(x_0x_1x_2) = \frac{x_0 - x_2}{\rho_2(x_0x_1) - \rho_2(x_1x_2)} + \rho_1(x_1), \tag{25}$$

$$\rho_4(x_0x_1x_2x_3) = \frac{x_0 - x_3}{\rho_3(x_0x_1x_2) - \rho_3(x_1x_2x_3)} + \rho_2(x_1x_2), \tag{26}$$

$$\vdots$$

$$\rho_{n+1}(x_0x_1\cdots x_n) = \frac{x_0 - x_n}{\rho_n(x_0x_1\cdots x_{n-1}) - \rho_n(x_1x_2\cdots x_n)} + \rho_{n-1}(x_1x_2\cdots x_{n-1}). \tag{27}$$

We remark that substitution of $x_k = k$ in the above equations gives the $\rho-$algorithm (16). Let us replace x_0 by x in eqs. (23)$-$(27). Then they are equivalent to the following identities in x;

$$\begin{aligned}
f(x) &= \rho_1(x), && (28)\\
\rho_1(x) &= \rho_1(x_1) + \frac{x - x_1}{\rho_2(xx_1)}, && (29)\\
\rho_2(xx_1) &= \rho_2(x_1x_2) + \frac{x - x_2}{\rho_3(xx_1x_2) - f(x_1)}, && (30)\\
\rho_3(xx_1x_2) &= \rho_3(x_1x_2x_3) + \frac{x - x_3}{\rho_4(xx_1x_2x_3) - \rho_2(x_1x_2)}, && (31)\\
&\vdots\\
\rho_{n+1}(xx_1x_2\cdots x_n) &= \rho_{n+1}(x_1x_2\cdots x_{n+1})\\
&\quad + \frac{x - x_n}{\rho_{n+2}(xx_1x_2\cdots x_{n+1}) - \rho_n(x_1x_2\cdots x_n)}. && (32)
\end{aligned}$$

We obtain the following continued fraction expansion for $f(x)$ from eqs. (28)$-$(32);

$$f(x) = \rho_1(x_1) + \cfrac{x - x_1}{\rho_2(x_1x_2) + \cfrac{x - x_2}{\rho_3(x_1x_2x_3) - \rho_1(x_1) + \cfrac{x - x_3}{\rho_4(x_1x_2x_3x_4) - \rho_2(x_1x_2) + \ddots}}}. \tag{33}$$

When we take $n-$th convergent of eq. (33), we obtain a rational function, which agrees in value with $f(x)$ at the points

$$x_1, x_2, \cdots, x_n.$$

Approximation of $f(x)$ by a rational function is called Thiele's interpolation. Let us rewrite

$$y = f(x),\ y_s = f(x_s),\ \rho_s = \rho_s(x_1x_2\cdots x_{s+1}) \tag{34}$$

for brevity. If we put $n-$th convergent of eq. (33) as $\dfrac{p_n(x)}{q_n(x)}$, we see inductively that $p_{2n+1}(x)$, $q_{2n+1}(x)$, and $p_{2n}(x)$ are polynomials in x of degree n while $q_{2n}(x)$ is a polynomial of degree $n-1$, and that these polynomials are written as

$$\begin{aligned}
p_{2n}(x) &= a_0 + a_1x + a_2x^2 + \cdots + a_{n-1}x^{n-1} + x^n, && (35)\\
q_{2n}(x) &= b_0 + b_1x + b_2x^2 + \cdots + b_{n-2}x^{n-2} + \rho_{2n}x^{n-1}, && (36)\\
p_{2n+1}(x) &= c_0 + c_1x + c_2x^2 + \cdots + c_{n-1}x^{n-1} + \rho_{2n+1}x^n, && (37)\\
q_{2n+1}(x) &= d_0 + d_1x + d_2x^2 + \cdots + d_{n-1}x^{n-1} + x^n. && (38)
\end{aligned}$$

Regarding

$$\frac{p_{2n}(x_s)}{q_{2n}(x_s)} = y_s \ (s = 1, 2, \cdots, 2n) \tag{39}$$

$$\frac{p_{2n+1}(x_s)}{q_{2n+1}(x_s)} = y_s \ (s = 1, 2, \cdots, 2n+1) \tag{40}$$

as simultaneous equations for $(a_0, a_1, \cdots, a_{n-1}, b_0, b_1, \cdots, b_{n-2}, \rho_{2n})$ and $(c_0, c_1, \cdots, c_{n-1}, d_0, d_1, \cdots, d_{n-1}, \rho_{2n+1})$, we see from the Cramer's formula that the quantities ρ_{2n} and ρ_{2n+1} are given by

$$\rho_{2n} = \rho_{2n}(x_1 x_2 \cdots x_{2n}) = \frac{|1, y_s, x_s, x_s y_s, x_s^2, \cdots, x_s^{n-2}, x_s^{n-2} y_s, x_s^{n-1}, x_s^n|}{|1, y_s, x_s, x_s y_s, x_s^2, \cdots, x_s^{n-2}, x_s^{n-2} y_s, x_s^{n-1}, x_s^{n-1} y_s|}$$

$$= \frac{\begin{vmatrix} 1 & y_1 & x_1 & x_1 y_1 & \cdots & x_1^{n-2} & x_1^{n-2} y_1 & x_1^{n-1} & x_1^n \\ 1 & y_2 & x_2 & x_2 y_2 & \cdots & x_2^{n-2} & x_2^{n-2} y_2 & x_2^{n-1} & x_2^n \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots \\ 1 & y_{2n} & x_{2n} & x_{2n} y_{2n} & \cdots & x_{2n}^{n-2} & x_{2n}^{n-2} y_{2n} & x_{2n}^{n-1} & x_{2n}^n \end{vmatrix}}{\begin{vmatrix} 1 & y_1 & x_1 & x_1 y_1 & \cdots & x_1^{n-2} & x_1^{n-2} y_1 & x_1^{n-1} & x_1^{n-1} y_1 \\ 1 & y_2 & x_2 & x_2 y_2 & \cdots & x_2^{n-2} & x_2^{n-2} y_2 & x_2^{n-1} & x_2^{n-1} y_2 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots \\ 1 & y_{2n} & x_{2n} & x_{2n} y_{2n} & \cdots & x_{2n}^{n-2} & x_{2n}^{n-2} y_{2n} & x_{2n}^{n-1} & x_{2n}^{n-1} y_{2n} \end{vmatrix}}, \tag{41}$$

$$\rho_{2n+1} = \rho_{2n+1}(x_1 x_2 \cdots x_{2n+1}) = \frac{|1, y_s, x_s, x_s y_s, x_s^2, \cdots, x_s^{n-1}, x_s^{n-1} y_s, x_s^n y_s|}{|1, y_s, x_s, x_s y_s, x_s^2, \cdots, x_s^{n-1}, x_s^{n-1} y_s, x_s^n|}$$

$$= \frac{\begin{vmatrix} 1 & y_1 & x_1 & x_1 y_1 & \cdots & x_1^{n-1} & x_1^{n-1} y_1 & x_1^n y_1 \\ 1 & y_2 & x_2 & x_2 y_2 & \cdots & x_2^{n-1} & x_2^{n-1} y_2 & x_2^n y_2 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 1 & y_{2n+1} & x_{2n+1} & x_{2n+1} y_{2n+1} & \cdots & x_{2n+1}^{n-1} & x_{2n+1}^{n-1} y_{2n+1} & x_{2n+1}^n y_{2n+1} \end{vmatrix}}{\begin{vmatrix} 1 & y_1 & x_1 & x_1 y_1 & \cdots & x_1^{n-1} & x_1^{n-1} y_1 & x_1^n \\ 1 & y_2 & x_2 & x_2 y_2 & \cdots & x_2^{n-1} & x_2^{n-1} y_2 & x_2^n \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 1 & y_{2n+1} & x_{2n+1} & x_{2n+1} y_{2n+1} & \cdots & x_{2n+1}^{n-1} & x_{2n+1}^{n-1} y_{2n+1} & x_{2n+1}^n \end{vmatrix}}. \tag{42}$$

Determinants in eqs. (41) and (42) become double Casorati determinants by changing their columns. It should be noted that eq. (20) is recovered by putting

$$x_k = k, \ y_k = S_k \tag{43}$$

in eqs. (41) and (42).

Next we extend the ρ−algorithm according to the notion of Thiele's interpolation. When we redefine $u^{(m)}(p; q)$ in eq. (20) by

$$u^{(m)}(p;q) = \det \begin{bmatrix} 1 & \sigma(m) & \cdots & \sigma(m)^{p-1} & S_m & \sigma(m)S_m & \cdots & \sigma(m)^{q-1}S_m \\ 1 & \sigma(m+1) & \cdots & \sigma(m+1)^{p-1} & S_{m+1} & \sigma(m+1)S_{m+1} & \cdots & \sigma(m+1)^{q-1}S_{m+1} \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 1 & \sigma(m+p+q-1) & \cdots & \sigma(m+p+q-1)^{p-1} & S_{m+p+q-1} & \sigma(m+p+q-1)S_{m+p+q-1} & \cdots & \sigma(m+p+q-1)^{q-1}S_{m+p+q-1} \end{bmatrix}, \tag{44}$$

we can construct extended version of the ρ-algorithm,

$$(x_{n+1}^{(m)} - x_{n-1}^{(m+1)})(x_n^{(m+1)} - x_n^{(m)}) = \sigma(n+m) - \sigma(m). \tag{45}$$

Since we obtain the ρ-algorithm by putting $\sigma(x) = x$ in eq. (45), we call eq. (45) "Thiele's ρ-algorithm". This algorithm accelerates sequences of the following form;

$$S_m \sim S_\infty + \frac{c_1}{\sigma(m)} + \frac{c_2}{(\sigma(m))^2} + \frac{c_3}{(\sigma(m))^3} + \cdots. \tag{46}$$

Let us apply Thiele's ρ-algorithm for two examples and compare its performance with the ρ-algorithm (16). First we consider a sequence

$$S_m = \sum_{k=1}^{m} \frac{1}{k^{3/2}} \to 2.61237534868\cdots, \tag{47}$$

whose asymptotic behavior is given by

$$S_m \sim S_\infty + \frac{c_1}{m^{1/2}} + \frac{c_2}{m^{3/2}} + \cdots. \tag{48}$$

In eq. (48), $c_i (i = 1, 2, \ldots)$ are constants. We put $\sigma(x) = x^{1/2}$ in eq. (45) and compare the result with the ρ-algorithm.

Next we consider the problem of evaluating

$$S_\infty = \int_0^1 g(x)\mathrm{d}x \tag{49}$$

by the trapezoidal rule. Define S_m by

$$S_m = \frac{1}{m}\left\{\frac{1}{2}g(0) + \sum_{k=1}^{m-1} g\left(\frac{k}{m}\right) + \frac{1}{2}g(1)\right\}, \tag{50}$$

If $g(x)$ is sufficiently smooth, an asymptotic behavior of S_m is given by

$$\begin{aligned} S_m &= S_\infty + \frac{d_1}{m^2} + \frac{d_2}{m^4} + \frac{d_3}{m^6} + \cdots, \\ & \quad d_1 = \frac{1}{12}\{g'(1) - g'(0)\},\ d_2 = -\frac{1}{720}\{g'''(1) - g'''(0)\}, \cdots. \end{aligned} \tag{51}$$

We put $g(x) = (0.05 + x)^{-1/2}$ in eq. (49) and apply Thiele's ρ–algorithm with $\sigma(x) = x^2$.

As one can see from Figures 4 and 5, Thiele's ρ–algorithm accelerates larger classes of sequences. We should select $\sigma(x)$ in eq. (45) appropriately according to asymptotics of a given sequence S_m.

§6 The PGR algorithms

In this section, we discuss the "PGR algorithms". First of all, we consider the most general form of the algorithm given by

$$(x_{n+1}^{(m)} - x_{n-1}^{(m+1)})(x_n^{(m+1)} - x_n^{(m)}) = z_n^{(m)}. \tag{52}$$

Papageorgiou et al. [4] applied the singularity confinement test to eq. (52). If we have $x_n^{(m+1)} = x_n^{(m)} + \delta$, then $x_{n+1}^{(m)}$ diverges as $z_n^{(m)}/\delta$. At the next iteration, we find

$$x_{n+2}^{(m)} = x_n^{(m+1)} + \left(1 - \frac{z_{n+1}^{(m)}}{z_n^{(m)}}\right)\delta + O(\delta^2),\ x_{n+2}^{(m-1)} = x_n^{(m)} + \frac{z_{n+1}^{(m-1)}}{z_n^{(m)}}\delta + O(\delta^2). \tag{53}$$

The singularity confinement condition, i.e. $x_{n+3}^{(m-1)}$ finite, is

$$z_n^{(m)} - z_{n+1}^{(m-1)} - z_{n+1}^{(m)} + z_{n+2}^{(m-1)} = 0. \tag{54}$$

The $\varepsilon(z_n^{(m)} = 1)$, $\rho(z_n^{(m)} = n)$, and Thiele's $\rho(z_n^{(m)} = \sigma(m+n) - \sigma(m))$ algorithms pass the above condition (54). We call the most generalized class of algorithms (52), where $z_n^{(m)}$ satisfies eq. (54), "the PGR algorithms". It is interesting to remark that Brezinski and Redivo Zaglia [16] have already proposed the algorithm (52) satisfying the singularity confinement condition (54), which they termed "the homographic invariance".

Two question arises; (1) what kind of integrable equations are the PGR algorithms associated with if they are considered as difference equations? (2) how is the performance of PGR algorithms as convergence accelerators?

As one solution to the first question, we have recently found that it is related to the discrete Painlevé equation of type I [17]. We consider a special case of the PGR algorithms,

$$(x_{n+1}^{(m)} - x_{n-1}^{(m+1)})(x_n^{(m+1)} - x_n^{(m)}) = n - m + C \ (C = \text{const}), \tag{55}$$

which passes the singurality confinement test. Through variable transformations,

$$k = n - m, \; l = m, \tag{56}$$

we have

$$(X(k+1,l) - X(k-2,l+1))(X(k-1,l) - X(k,l)) = k + C. \tag{57}$$

Elimination of the dependence of the variable l in eq. (57) gives the following equation;

$$X(k+1) - X(k-2) = \frac{-k-C}{X(k) - X(k-1)}. \tag{58}$$

Through dependent variable transformation,

$$Y(k) = X(k) - X(k-1), \tag{59}$$

we have

$$Y(k+1) + Y(k) + Y(k-1) = \frac{-k-C}{Y(k)} \tag{60}$$

from eq. (58). The equation (60) is nothing but the discrete Painlevé equation of type I. It is noted, however, the algorithm (55) does not accelerate convergence of sequences well.

Let us go to the second question, i.e. acceleration performance of the PGR algorithms. Intuitively speaking, most of the PGR algorithms do not well accelerate convergence as far as we have tested them. Among them, however, when we take

$$z_n^{(m)} = \frac{1}{m+n+1}, \tag{61}$$

the algorithm accelerates both alternatively and rationally decaying sequences (See Figures 6 – 8).

§7 Concluding Remarks

We have shown there being a strong tie between convergence acceleration algorithms and discrete soliton equations. It is a future problem to clarify how two different notions, acceleration and integrability, are associated with each other. In other words, we should consider whether we can construct new convergence acceleration algorithms from the other discrete soliton equations[2] and what kind of equations the other algorithms correspond to. The solution of these problems will shed a new light on the study of discrete integrable systems and numerical analysis.

[2] Actually, Papageorgiou et al. have proposed a new algorithm based on the discrete modified KdV equation.

It is a pleasure to thank Dr. Yasuhiro Ohta at Hiroshima University and Dr. Tetsu Yajima at Tokyo University for many fruitful discussions on the subject. We would like to thank Prof. Naoki Osada at Nagasaki Institute of Applied Science for valuable pieces of information on the $\rho-$algorithm and the singularity confinement.

Appendix A Numerical Results

We here present numerical examples. In Figure 4 the $\rho-$algorithm and Thiele's $\rho-$algorithms are applied to the sequence $S_m = \sum_{k=1}^{m} \frac{1}{k^{3/2}}$. These two algorithms are also applied to the numerical integration for $\int_0^1 \frac{1}{\sqrt{0.05+x}}\mathrm{d}x$ in Figure 5. We also employed $\varepsilon-$algorithm, $\rho-$algorithm, and PGR algorithm with $z_n^{(m)} = \frac{1}{m+n+1}$ to accelerate the series,

$$1-\frac{1}{2}+\cdots+\frac{(-1)^{m-1}}{m}+\cdots \to \log 2 (= 0.69314718\cdots), \tag{62}$$

$$1+\frac{1}{2^2}+\cdots+\frac{1}{m^2}+\cdots \to \frac{\pi^2}{6}(= 1.6449336\cdots), \tag{63}$$

$$\frac{1}{2\cdot 5}+\frac{1\cdot 3}{2\cdot 4\cdot 9}+\cdots+\frac{(2m-1)!!}{(2m)!!(4m+1)}+\cdots \to \int_0^1 \frac{\mathrm{d}x}{\sqrt{1-x^4}} - 1(= 0.31102877\cdots), \tag{64}$$

numerical results of which are given in Figures 6 – 8.

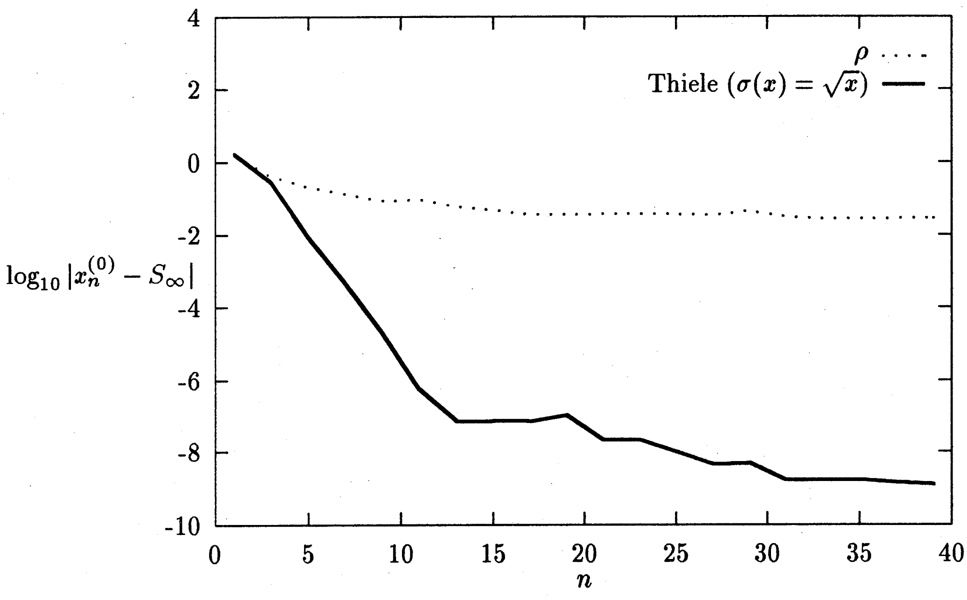

Figure 4: Acceleration methods for $\sum_{k=1}^{\infty} \frac{1}{k^{3/2}}$

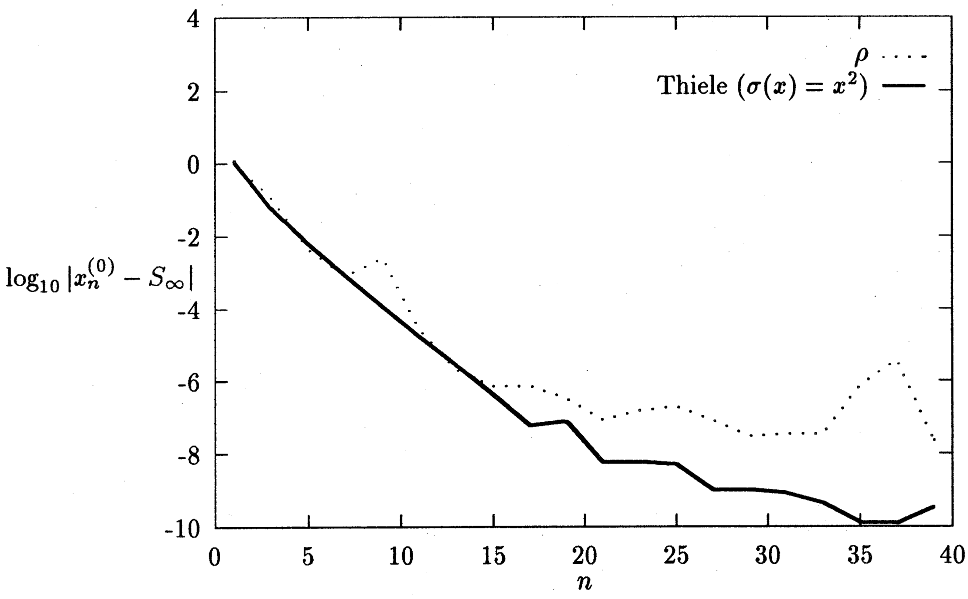

Figure 5: Acceleration methods for $\int_0^1 \frac{1}{(0.05+t)^{1/2}} \mathrm{d}t$

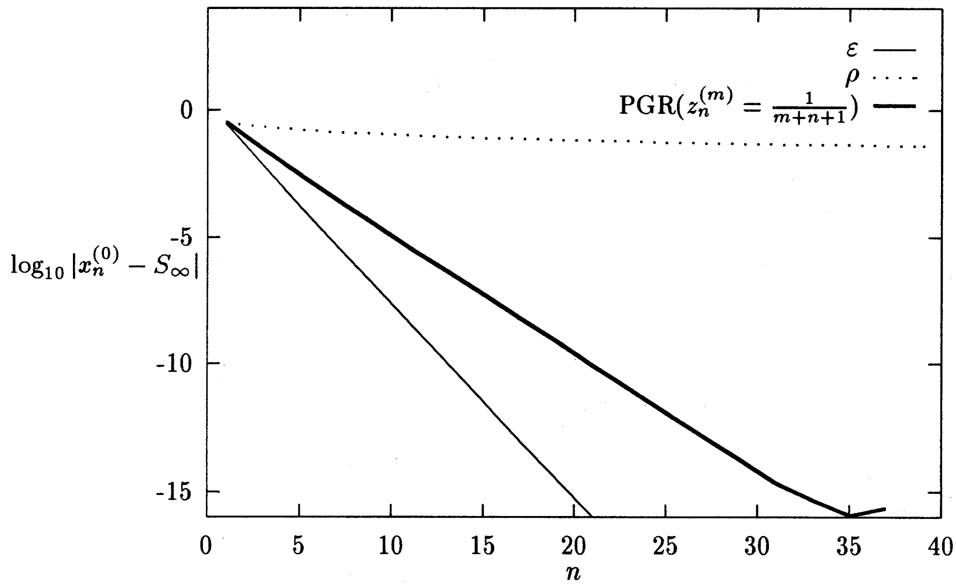

Figure 6: Acceleration methods for $\sum_{k=1}^{\infty} \frac{(-1)^{k-1}}{k}$

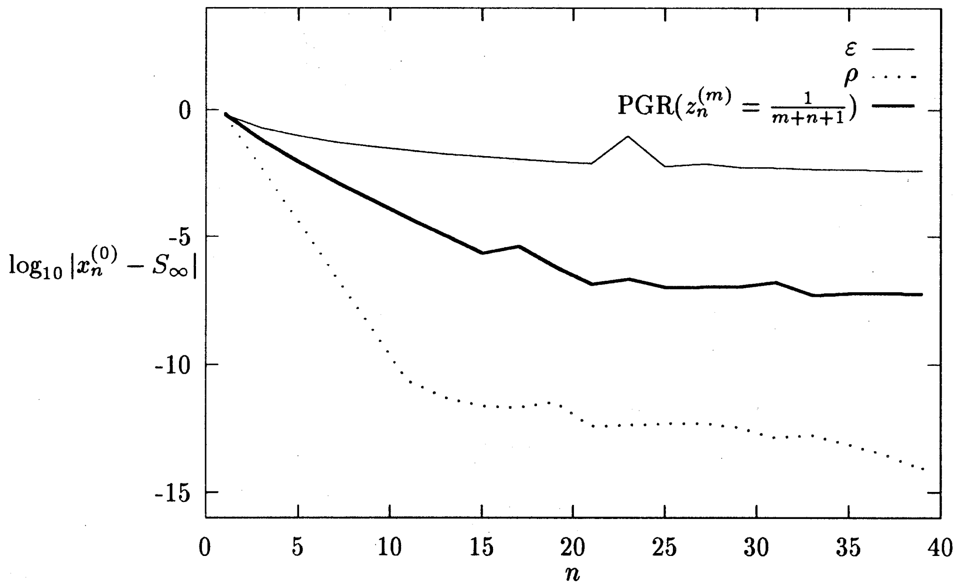

Figure 7: Acceleration methods for $\sum_{k=1}^{\infty} \frac{1}{k^2}$

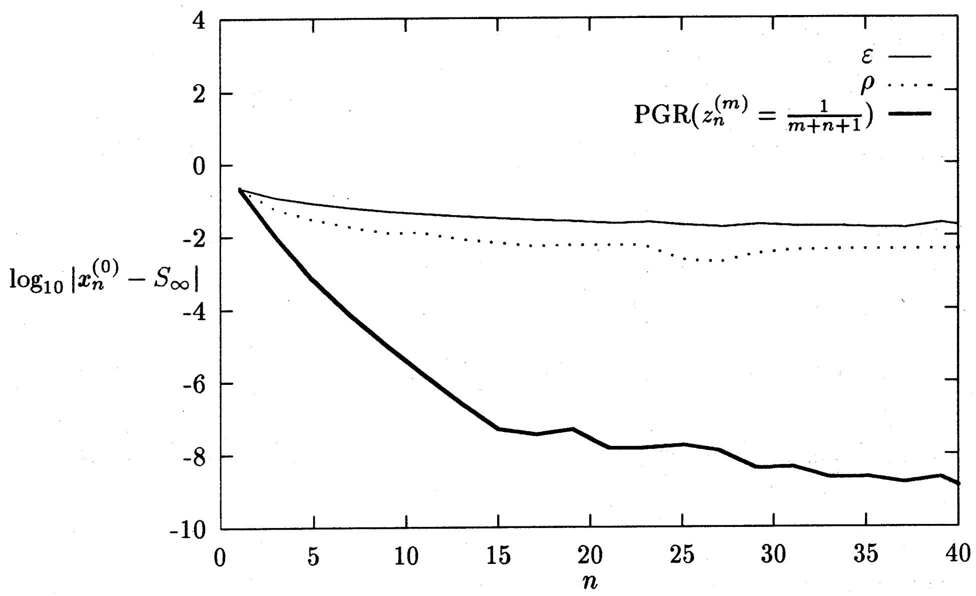

Figure 8: Acceleration methods for $\displaystyle\sum_{k=1}^{\infty} \frac{(2k-1)!!}{(2k)!!(4k+1)}$

References

[1] R. Hirota, J. Phys. Soc. Jpn. **43** (1977) 1424, **50** (1981) 3785.

[2] W. W. Symes, Physica 4D (1982) 275.

[3] R. Hirota, S. Tsujimoto, and T. Imai, RIMS Kokyuroku **822** (1992) 144.

[4] V. Papageorgiou, B. Grammaticos, and A. Ramani, Phys. Lett. A **179** (1993) 111.

[5] P. Wynn, Math. Tables Aids Comput. **10** (1956) 91.

[6] F. L. Bauer, in: On Numerical Approximation, ed. R. E. Langer (University of Wisconsin Press, Madison, 1959) p.361.

[7] T. N. Thiele, Interpolationsrechnung (Taubner, Leibzig, 1909).

[8] P. Wynn, Proc. Cambridge Philos. Soc. **52** (1956) 663.

[9] R. Bulirsh and J. Stoer, Numer. Math. **8** (1966) 1.

[10] D. Levin, Int. J. Comput. Math. **B3** (1973) 371.

[11] C. Brezinski, Lecture notes in mathematics, Vol. 584, Accélération de la convergence and analyse numérique (Springer, Berlin, 1977).

[12] S. Maxon and J. Viecelli, Phys. Fluids. **17** (1974) 1614.

[13] D. Shanks, J. Math. Phys. **34** (1955) 1.

[14] H. W. Capel, F. W. Nijhoff, and V. G. Papageorgiou, Phys. Lett. A **155** (1991) 377.

[15] D. A. Smith and W. F. Ford, SIAM J. Numer. Anal. **16** (1979) 223.

[16] C. Brezinski and M. Redivo Zaglia, Extrapolation Methods. Theory and Practice (North-Holland, Amsterdam, 1991).

[17] A. Ramani, B. Grammaticos, and J. Hietarinta, Phys. Rev. Lett. **67** (1991) 1829.